# 离散型流匹配模型

本文是接着上一次学习的CTMC模型继续学习离散型流匹配模型，主要是读Flow Matching Guide and Code[[1]](#footnote-21)第7章的笔记。并非完全Flow Matching Guide and Code的翻译，挑了一些重要的内容并且加入了我自己的的理解。

要构建一个离散型流匹配模型（Discrete Flow Matching，DFM），分为三个步骤：

1. 类比连续型FM，定义一个概率路径，连接原始分布的pmf和目标分布的pmf。pmf就是概率质量函数。
2. 构建一个CTMC模型，，该CTMC的速度场生成出第1步中的概率路径，是可学习的参数。
3. 训练，使之最小化Bregman散度，Bregman散度就是DFM的损失函数。

后面我就把离散型流匹配模型简称为DFM，连续型流匹配模型简称为FM。

## 数据和数据的成对

像FM里面一样，定义源分布的pmf为，目标分布的pmf为。源分布的随机变量（Random Variable，RV）记为，目标分布的随机变量记为。，就是CTMC模型里面提到的状态空间。

考虑到我们一般把和放在一起讨论，所以需要找个方式表示和的成对分布：，就是当成两个独立的RV，用二者的pmf的乘积表示的联合分布。也可以用单独的符号表示和的联合分布。

举个例子理解，如果对文本生成任务，源分布就是先验分布，一般取：

1. 上的均匀分布
2. 加入一个mask的特殊token 到vocabulary 里面，即。

按第2种方法，此时得到，先验分布就是。若，则。

## 离散概率路径

FM[[2]](#footnote-24)在研究边缘概率路径和生成的边缘向量场时遇到困难（图像在高维空间中造成[[3]](#footnote-25)中的公式6和8的积分不可解，因此没办法得到和的解析式），所以转而求解条件向量场和条件概率路径。仿照这个办法，DFM也可以按如下方式算边缘概率路径（marginal probability path）：

其中是作为条件的随机变量，是任意空间。上式是离散版的全概率公式。需要指出的是，**条件概率路径必须满足边界约束和，**这样才能构建源分布和目标分布之间的关系。

## 边缘化技巧

我们在公式1中已经给出了边缘概率路径和条件概率路径之间的关系，类比在FM里面做的事情，对DFM，我们也想知道条件速度场和边缘条件速度场之间的关系。

为了研究这个问题，可以从FM中获得经验。FM中，如果条件向量场能生成条件概率路径，那边缘向量场满足：

$$u\_t(x) = \int u\_t(x\mid z)\cfrac{p\_{t\mid Z}(x\mid z)p\_Z(z)}{p\_t(x)}\mathrm{d} z
=\int u\_t(x\mid z) p\_{Z\mid t}(z\mid x) \mathrm{d}z
\qquad\text{(2)}$$

公式2可以参考Flow matching for generative modeling[[4]](#footnote-27)里面的公式8，或者Flow Matching Guide and Code[[5]](#footnote-28)里面的公式4.12。

类比公式2，是否DFM也存在如下关系：当条件速度场生成条件概率路径时，边缘速度场满足

$$u\_t(y,x) = \sum\_z u\_t(y,x \mid z) \cfrac{p\_{t\mid Z}(x\mid z)p\_Z(z)}{p\_t(x)}
=\sum\_z u\_t(y,x\mid z) p\_{Z\mid t}(z\mid x) \qquad\text{(3)}$$

答案是肯定的。接下来详细表述和证明这个问题。

**假设** 。。对所有和，。

**定理** （离散边缘化技巧）当上述假设满足时，如果生成，那么对，边缘向量场（公式3）生成边缘概率路径（公式1）。

**证明**

$$\begin{aligned}
\cfrac{\mathrm{d}}{\mathrm{d}t}p\_t(y) &\overset{(1)}{=} \cfrac{\mathrm{d}}{\mathrm{d}t}\left[
\sum\_z{
p\_{t\mid Z}\left(y\mid z\right)p\_{Z}(z)
}\right]\\
&\overset{(2)}{=} \sum\_z{
\cfrac{\mathrm{d}}{\mathrm{d}t}\left[
p\_{t\mid Z}\left(y\mid z\right)p\_{Z}(z)
\right]
}\\
&\overset{(3)}{=} \sum\_z{
\cfrac{\mathrm{d}}{\mathrm{d}t}\left[
p\_{t\mid Z}\left(y\mid z\right)
\right]p\_{Z}(z)
}\\
&\overset{(4)}{=} \sum\_z {\sum\_x \left[{
u\_t(y,x \mid z) p\_{t\mid Z}(x\mid z)
} \right] p\_Z(z)}\\
&\overset{(5)}{=} \sum\_x {\sum\_z \left[{
u\_t(y,x \mid z) \cfrac{p\_{t\mid Z}(x\mid z)p\_Z(z)}{p\_t(x)}
} \right] p\_t(x)}\\
&\overset{(6)}{=} \sum\_x {\overbrace{\sum\_z \left[{
u\_t(y,x \mid z) p\_{Z\mid t}(z\mid x)
} \right]}^{u\_t(y,x)}
p\_t(x)}
\end{aligned}
\qquad\text{(4)}$$

第（4）步是因为根据条件“生成”，我们在CTMC模型中讲到的离散质量守恒Discrete Mass Conservation（Discrete Mass Conservation的假设已经满足了，就是上文给出的假设），可知：

* 生成
* 和满足Kolmogorov方程，且 满足速率条件（rate conditions）。

二者等价，故直接带入Kolmogorov方程可得第（4）步，只是加了个条件而已。

第（5）是将拿到内层求和，并在内层求和的分母上添加一个，此时还需要再乘，可以用分配律把它拿出内层求和。

第（6）步是贝叶斯公式。

最终得出的等式比对一下Kolmogorov方程，可见和满足Kolmogorov方程。

然后，满足速率条件，因为满足速率条件，即上文蓝色字的部分。

关于为什么满足速率条件，就满足速率条件的证明：

如果满足速率条件，那么 且

根据，因为显然不小于0，故。

所以也满足速率条件。

在再次应用Discrete Mass Conservation之前，还得确定一下假设是否满足，因为且，故，。

关于为什么且，就有。我自己的理解如下：

因为，在上连续，和无关，所以在连续。的一阶导在上连续，和无关，，故也在上连续。

因为$u\_t(y,x)=\sum\_z u\_t(y,x \mid z) p\_{Z\mid t}(z\mid x)= \sum\_z u\_t(y,x \mid z) \cfrac{p\_{t\mid Z}(x\mid z)p\_Z(z)}{p\_t(x)}$，在时，和都连续，和无关，根据假设，同时在连续，所以在连续。从这里能看到假设为什么一定要。

从上面这些解释就知道，。

所以假设也满足，可以应用Discrete Mass Conservation，得到**生成**。证明完毕。

**补充** 关于时，的假设是可以满足的。因为我们可以使用，其中是均匀分布，是大于0的很小的值。我的理解就是通过稍微加入一点服从均匀分布的噪声，让上，都有概率。

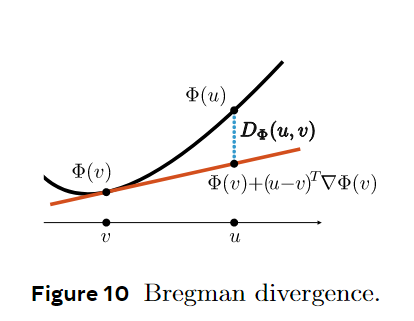
## DFM损失

像FM里面一样，我们可以用一个神经网络来学习速度场，其中是可学习的参数。类比FM，可以构建DFM的损失：

这里，，满足速率条件。也就是说，，且：

是凸集合。是Bregman散度，其使用凸函数定义。

Bregman的具体计算方式如下图和下式所示：



Bregman散度有一些关键的性质，在FM和DFM里面最关键的是它对第二个参数的梯度有仿射不变性（affine invariant）[[6]](#footnote-31)：

因为仿射不变性，可以得到。

关于Bregman散度的详细信息，在知乎上已经有很好的回答了，可以参考：<https://www.zhihu.com/question/22426561>。

就像在FM里面一样，我们更关注条件DFM（Conditional Discrete Flow Matching，CDFM）的损失函数形式，因为我们想仿照FM里面一样用神经网络预测条件速度场，如Flow matching for generative modeling[[7]](#footnote-33)里面的公式9所示。那么CDFM的损失就是：

是目标条件速度场，是模型学习的条件速度场。

像FM里面一样，我们也能说明公式6和公式10的梯度一样：

**定理** DFM和CDFM损失的梯度相同：

使得CDFM损失最小的是：

我参照[[8]](#footnote-34)以及一些自己的理解给了**证明**。

先看公式11的证明：

再看公式12的证明：

代入公式12到中，得到

按照Bregman散度的定义，可知，是一个全局最小值。所以使得最小的满足公式12。

在Flow Matching Guide and Code[[9]](#footnote-35)21页最上面（公式4.25和4.26）还有一个更一般化的证明。

这一节先到这里吧。

1. Lipman Y, Havasi M, Holderrieth P, et al. Flow Matching Guide and Code[J]. arXiv preprint arXiv:2412.06264, 2024. [↑](#footnote-ref-21)
2. Lipman Y, Chen R T Q, Ben-Hamu H, et al. Flow matching for generative modeling[J]. arXiv preprint arXiv:2210.02747, 2022. [↑](#footnote-ref-24)
3. Lipman Y, Chen R T Q, Ben-Hamu H, et al. Flow matching for generative modeling[J]. arXiv preprint arXiv:2210.02747, 2022. [↑](#footnote-ref-25)
4. Lipman Y, Chen R T Q, Ben-Hamu H, et al. Flow matching for generative modeling[J]. arXiv preprint arXiv:2210.02747, 2022. [↑](#footnote-ref-27)
5. Lipman Y, Havasi M, Holderrieth P, et al. Flow Matching Guide and Code[J]. arXiv preprint arXiv:2412.06264, 2024. [↑](#footnote-ref-28)
6. Holderrieth P, Havasi M, Yim J, et al. Generator Matching: Generative modeling with arbitrary Markov processes[J]. arXiv preprint arXiv:2410.20587, 2024. [↑](#footnote-ref-31)
7. Lipman Y, Chen R T Q, Ben-Hamu H, et al. Flow matching for generative modeling[J]. arXiv preprint arXiv:2210.02747, 2022. [↑](#footnote-ref-33)
8. Lipman Y, Havasi M, Holderrieth P, et al. Flow Matching Guide and Code[J]. arXiv preprint arXiv:2412.06264, 2024. [↑](#footnote-ref-34)
9. Lipman Y, Havasi M, Holderrieth P, et al. Flow Matching Guide and Code[J]. arXiv preprint arXiv:2412.06264, 2024. [↑](#footnote-ref-35)